

实验三 综合案例（1）

一、 实验目的

- 1、 掌握类定义
- 2、 掌握类设计

二、 实验内容

1、 PTA 作业《C++语言程序设计工程实践：面向对象基础》

2、 使用 Visual Studio 环境来编辑、编译和运行下列程序。

一个人可以有多个活期储蓄账户，一个活期储蓄账户包括账号(id)、余额(balance)、结算年利率(rate)、等信息,还包括显示账户信息(show)存款(deposit)、取款(withdraw)、账户利息(settle)等操作。为此,设计一个类 SavingsAccount,将 id,balance,rate 均作为其成员函数。无论是存款、取款还是结算利息都需要修改当前的余额并且将余额的变动输出，这些公共操作由私有成员函数 record 来执行。由于账户的余额是不断变化的，因此不能通过余额和年利率相乘的方法来计算年利，需要将一年当中每天的余额累积起来再除以一年的总天数，得到一个日均余额，再乘以年利率。为了计算余额的按日累积值，SavingsAccount 引入了私有数据成员 lastDate,accumulation 和私有成员函数 accumulate。lastDate 用来存储上一次余额变动的日期, accumulation 用来存储上次计算利息以后直到最近一次余额变动时余额按日累加的值,accumulate 成员函数用来计算截至指定日期的账户余额按日累积值。这样，当余额变动时，需要做的是将变动前的余额与该余额所持续的天数相乘，累加到 accumulation 中，再修改 lastDate。

//代码内容 1：类的定义

```
#include <iostream>
#include <cmath>
using namespace std;
class SavingsAccount { //储蓄账户类
private:
    int id;           //账号
    double balance;   //余额
    double rate;      //存款的年利率
    int lastDate;     //上次变更余额的时期
    double accumulation; //余额按日累加之和
    //记录一笔帐， date 为日期， amount 为金额， desc 为说明
    void record(int date, double amount);
    //获得到指定日期为止的存款金额按日累积值
    double accumulate(int date) const {
```

```

        return accumulation + balance * (date - lastDate);
    }
public:
    //构造函数
    SavingsAccount(int date, int id, double rate);
    int getId() { return id; }
    double getBalance() { return balance; }
    double getRate() { return rate; }
    void deposit(int date, double amount); //存入现金
    void withdraw(int date, double amount); //取出现金
    //结算利息，每年 1 月 1 日调用一次该函数
    void settle(int date);
    //显示账户信息
    void show();
};

//SavingsAccount 类相关成员函数的实现
SavingsAccount::SavingsAccount(int date, int id, double rate)
    : id(id), balance(0), rate(rate), lastDate(date), accumulation(0) {
    cout << date << "\t#" << id << " is created" << endl;
}

void SavingsAccount::record(int date, double amount) {
    accumulation = accumulate(date);
    lastDate = date;
    amount = floor(amount * 100 + 0.5) / 100; //保留小数点后两位
    balance += amount;
    cout << date << "\t#" << id << "\t" << amount << "\t" << balance <<
endl;
}

void SavingsAccount::deposit(int date, double amount) {
    record(date, amount);
}

void SavingsAccount::withdraw(int date, double amount) {
    if (amount > getBalance())
        cout << "Error: not enough money" << endl;
    else
        record(date, -amount);
}

void SavingsAccount::settle(int date) {
    double interest = accumulate(date) * rate / 365; //计算年息
    if (interest != 0)
        record(date, interest);
    accumulation = 0;
}

void SavingsAccount::show() {

```

```

        cout << "#" << id << "\tBalance: " << balance;
    }
int main() {
    //建立几个账户
    SavingsAccount sa0(1, 21325302, 0.015);
    SavingsAccount sa1(1, 58320212, 0.015);
    //几笔账目
    sa0.deposit(5, 5000);
    sa1.deposit(25, 10000);
    sa0.deposit(45, 5500);
    sa1.withdraw(60, 4000);
    //开户后第 90 天到了银行的计息日， 结算所有账户的年息
    sa0.settle(90);
    sa1.settle(90);
    //输出各个账户信息
    sa0.show(); cout << endl;
    sa1.show(); cout << endl;
    return 0;
}

```

三、 实验要求

- 1、 完成部分 PTA 作业《C++语言程序设计工程实践：面向对象基础》的实验题目。
- 2、 成功在 Visual Studio 2019 里面运行上述代码。